MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, **2018/2019**

HPB2021 – BIOINFORMATICS PROGRAMMING II

(All sections / Groups)

19 OCTOBER 2018
03:00 to 05:00 PM
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 4 pages including cover page.

2. Attempt all FOUR questions. All questions carry equal marks (10), and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

QUESTION 1 [10 marks]

Write down the expected output of the following codes.

a)Python

[1 mark]

```
i.def addition(x, y):
   print x + y,
   print x * y

 addition("2", 3)
 addition("2", "3")
```

[0.5 mark]

```
ii. def what(n):
      if n == 0:
        return 0
      else:
        return n + what(n-1)
print what(5)
```

[1 mark]

```
iii. s = "Alabama and Alaska"
      s[ : 7]
      s[ 1 : 7 : 2]
      s[ 2 : 7 :-1]
      s[ : : -1]
```

[1 mark]

```
iv. text = raw input("Input text: ")
      vowels = ['a', 'e', 'i', 'o', 'u', 'y' ]
      for c in text.lower():
            if not c.isalpha():
                  continue
            if c in vowels:
                  print c,
                  continue
            print "%sO%s"%(c, c),
      Input text: The light was yellow, sir.
```

[1 mark]

```
v. m = [ ['c'] * 3 ] * 3
    print m
    m[0][2] = 7
    print m
```

[0.5 mark]

```
vi. x= “Hi “
     y= “Bye ”
    x, y = y, x
    print x+y
```

[1 mark]

```
vii.import re
for test_string in ['555-1212', 'ILL-
EGAL']:
   if re.match(r'^\d{3}-\d{4}$',
test_string):
       print test_string, 'is a valid US local
phone number'
   else:
           print test_string, 'rejected'
```

[0.5 mark]

```
viii.
for y in range(25, 15, -1):
   if not y%3:
         print y,
```

b) R Programming

[1 mark]

```
i. > x <- c(1:10)
   > x[(x>8) | (x<5)]
   > x > 8 | x < 5
```

[1mark]

```
ii.>smoker=matrix(c(70,50,65,140),
ncol=2,byro=FALSE)
> rownames(smoker)=c("male","female")
> colnames(smoker)=c("smoke","nosmoke")
> smoker= as.table(smokers)
> smokers
```

[1 mark]

```
iii.> seq(3,30,length=10)
```

[0.5 mark]

```
iv. > x=1:6
> y=cut(x,3)
>y
```

Continued…

QUESTION 2 [10 marks]

a) Define a function "validate_base_sequence()" to check whether a given DNA sequence only contains "A", "T", "C", and "G" and a given RNA only contains "A", "U", "C", "G". This function receives two arguments:

i. The base sequence for validation

ii. A boolean flag "RNAflag" (initially False) to show if the base sequence is DNA or RNA [3 marks]

The function return is True (all the bases in sequence are validated for RNA or DNA) or False (otherwise). The output for the function calls is as the following:

```
#>>> validate_base_sequence('ATCG', False)
#True
#>>> validate_base_sequence('ATCG', True)
#False
#>>> validate_base_sequence('AUCG', True)
#True
#>>> validate_base_sequence('ATCG')
#True
```

b) Write a function "generate_triples()" to return a list containing every three-character combination of the unique characters in its argument parameter (string parameter). It first converts the string argument to a set in order to eliminate any duplicates. This function can be used to generate a list of strings representing all the codons for a given collection of 'TCAG' or 'UCAG'. Some sample function calls can generate the following outputs [4 marks]

```
A string of any length:
>>> generate_triples('01')
['111', '110', '101', '100', '011', '010', '001', '000']
A string with duplicate charcters
>>> generate_triples('AATTCG') #duplicates discarded and four bases are
"ATCG"
['AAA', 'AAT', 'ATT',.... All possible codons]]
```

c) Write a function "read_fasta_seq()" that receives the file name as an argument and returns the sequence read from the file. Each time (implement a loop) it reads a line, it checks whether the line is not empty (to indicate end of the file). If the line is not empty the function checks if the line starts with ">"; if yes, discard the line and if not, append the line at the end of long string sequence variable. [3 marks]

Continued...

QUESTION 3 [10 marks]

a) Given "x<-matrix(c(2,-3,5,-1),nrow=2,byrow=T)" and "y<-matrix(c(4,2,-5,1,2,3),nrow=3,byrow=T)", write the output when:

```
>x
>y
> y%*%x
```

[4 marks]

b) A microarray genes expression values are tabulated in a comma delimited csv file, namely genedata.csv. Write an R code to import the content of the comma delimited csv file. There are headers in the file. Then, list any available packages or libraries that can be used to plot heatmap based on the extracted data.

[2 marks]

c) Write an R program to read a sequence from the user. The program counts the number of bases, "A","T","C", and "G" prints the length of the sequence, bases, and the number of occurrences. [4 marks]

QUESTION 4 [10 marks]

a) Read all the sequences in a GenBank file "example.gbk" using BioPython. Display the IDs and the sequences. Store the IDs in idList[], and the sequences in seqList[] [5 marks]

b) Write BioPython code to create a Seq object of DNA using the IUPAC.unambiguous_dna alphabet find the reverse complement of the DNA sequence and to transcribe from the DNA to a mRNA sequence.

[5 marks]

End of Paper